

**IN THE UNITED STATES PATENT AND TRADEMARK OFFICE  
BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES  
Attorney Docket No.003797.00622**

In re the Patent Application of: Manu, et al.      )  
Application No. 10/700,178                              )      Group Art Unit: 2857  
  )  
Filed: November 3, 2003                              )  
For: Flexible Variable and Execution Matrix      )      Examiner: Tsai  
  )  
  )      Confirmation No. 3590  
  )

**BRIEF ON APPEAL**

Mail Stop: Appeal Brief-Patents  
Commissioner of Patents  
P.O. Box 1450  
Alexandria, VA 22313-1450

This is an appeal brief in accordance with 37 CFR §1.192 filed in support of applicant's October 13, 2005, Notice of Appeal. Appeal is taken from the office action mailed July 15, 2005. Please charge any necessary fees in connection with this appeal brief to our Deposit Account No. 19-0733.

**I.      REAL PARTY IN INTEREST**

The owner of this application, and the real party in interest, is Microsoft Corporation.

**II.     RELATED APPEALS AND INTERFERENCES**

There are no related appeals and interferences.

**III.    STATUS OF CLAIMS**

Claims 1-17 remain in the application. All of the pending claims are shown in the attached appendix.

**IV. STATUS OF AMENDMENTS**

There are no amendments subsequent to the final office action dated June 20, 2005, and all prior amendments have been entered.

**V. SUMMARY OF CLAIMED SUBJECT MATTER**

In making reference herein to various portions of the specification and drawings in order to explain the claimed invention (as required by 37 CFR §41.37(c)(1)(v)), Applicant does not intend to limit the claims. All references to the specification and drawings are illustrative unless otherwise explicitly stated.

The claimed subject matter is directed to the field of hardware and software testing. (Page 1, Paragraph 1, lines 1-3). Aspects of the invention provide improved systems and methods for selecting parameter values and combinations of parameter values to use when testing software modules. (Page 2, Paragraph 5, lines 1-3). The testing of software modules typically includes selecting parameter values and parameter value combinations. (Page 1, Paragraph 2, lines 3-4). The combinations of parameter values are then applied to the software module and the resulting output is analyzed. (Page 1, Paragraph 2, lines 3-4). The selection of parameter values and combinations of parameter values is critical to ensure that a software module is operating properly. (Page 1, Paragraph 3, lines 1-2). It is generally desirable to test the values and combinations of values that are most likely to occur during the operation of the software module and existing systems and methods for selecting parameter values and combinations of parameter values can be time consuming and error prone. (Page 1, Paragraph 3, lines 2-5).

Independent claim 1 is directed to a “method of generating a list of parameter value combinations to test.” (Page 7, Paragraph 24, lines 1-2; Figure 6, step 602). Claim 1 includes

four elements. The first element is “(a) providing to a user a graphical user interface that includes at least two adjustable probability curves that allow the user to graphically indicate the importance of values of at least first and second parameters.” (Page 7, Paragraph 24, lines 2-5). Figure 2 shows an exemplary graphical user interface. A user may use icons to adjust the shape of the curve shown in Figure 2 to correspond to the importance or interest of parameter values. (Page 6, Paragraph 20, lines 5-10). Figure 3 shows an exemplary probability curve in which parameter values of 2, 4 and 6 are of high interest and the parameter value of 5 is of relatively low interest. (Page 6, Paragraph 22, lines 1-4). The relative interest of a particular parameter value may be the result of numerous factors determined by a user. For example, a high interest parameter value may be a value that is likely to occur when a software module is in operation or a critical value. (Pages 6-7, Paragraph 22, lines 4-7).

The next element of claim 1 is “(b) converting the probability curves into probability functions.” (Page 7, Paragraph 25, line 1; Figure 6, step 604). Converting the probability curves into probability functions may include performing curve fitting, such as polynomial curve fitting. (Pages 7, Paragraph 25, lines 1-2).

The next element of claim 1 is “(c) combining the probability functions into a combination function.” (Page 7, Paragraph 25, lines 3-4; Figure 6, step 606). On page 8 the specification provides the exemplary formula for performing this process:

(equation 1)

$$\sum_{i=1}^n (2 * P(x_i) * \text{Max}(P(x_i)) - P(x_i)^2)$$

In an alternative embodiment the combination function is equal to the product of the probability functions. (Page 8, Paragraph 26, lines 2-4).

The final element in claim 1 is “(d) selecting parameter value combinations that result in the combination function exceeding a predetermined probability value.” (Page 8, Paragraph 27, lines 1-2; Figure 6, step 608).

Independent claim 14 is directed to a “method of generating a list of parameter values to test.” (Page 7, Paragraph 24, lines 1-2; Figure 6, step 602). Claim 14 includes three elements. The first element is “(a) providing to a user a graphical user interface that includes an adjustable probability curve that allows the user to graphically indicate the importance of values of a parameter.” (Page 7, Paragraph 24, lines 2-5). Figure 2 shows an exemplary graphical user interface. A user may use icons to adjust the shape of the curve shown in Figure 2 to correspond to the importance or interest of parameter values. (Page 6, Paragraph 20, lines 5-10). Figure 3 shows an exemplary probability curve in which parameter values of 2, 4 and 6 are of high interest and the parameter value of 5 is of relatively low interest. (Page 6, Paragraph 22, lines 1-4). The relative interest of a particular parameter value may be the result of numerous factors determined by a user. For example, a high interest parameter value may be a value that is likely to occur when a software module is in operation or a critical value. (Pages 6-7, Paragraph 22, lines 4-7).

The next element in claim 14 is “(b) converting the probability curve into a probability function.” (Page 7, Paragraph 25, line 1; Figure 6, step 604). Converting the probability curve into a probability function may include performing curve fitting, such as polynomial curve fitting. (Pages 7, Paragraph 25, lines 1-2).

The last element in claim 14 is “(c) selecting parameter values that result in the probability function exceeding a predetermined probability value.” (Page 8, Paragraph 27, lines 1-2; Page 8, Equation 1 when n=1).

The last independent claim is claim 16. Claim 16 is drawn to a “method of testing a software module with parameter combinations” and includes four elements. Figure 7 illustrates an exemplary graphical user interface that may be used with the method of claim 16. The first element in claim 16 is “(a) displaying in a first region of the display a list of parameter combinations.” The graphical user interface in Figure 7 includes a first region 702 that includes an execution matrix of parameter combinations to test. (Page 8, Paragraph 29, lines 2-3; Figure 7).

The next element is “(b) displaying in a second region of the display an input icon.” Figure 7 shows an input icon 704 displayed as part of the graphical user interface. (Pages 8-9, Paragraph 29, lines 3-4; Figure 7).

The next element in claim 16 is “(c) receiving an indication from a user to drag at least one of the parameter combinations to the input icon.” A user may test a particular parameter value combination by using a pointing device to select that combination from the execution matrix and dragging that combination to input icon 704. (Page 9, Paragraph 29, lines 5-7; Page 4, Paragraph 16, lines 3-5).

The last element in claim 16 is “(d) in response to (c) displaying in a third region of the display an output of the software module.” After the software module has operated on the parameter value combination, the results of the operation may be displayed in an output region 708. (Page 9, Paragraph 29, lines 7-8; Figure 7).

## VI. GROUNDS OF REJECTION TO BE REVIEWED ON APPEAL

- 1) Claims 1-17 stand rejected under 35 USC §101 as being directed to non-statutory subject matter.

- 2) Claims 1-17 stand rejected under 35 USC §112, first paragraph as failing to comply with the written description requirement.
- 3) Claims 1-15 stand rejected under 35 USC §112, second paragraph as being indefinite.

## VII. ARGUMENT

### A. Claims 1-17 Contain Statutory Subject Matter

#### Claims 1-15

In the Office Action of July 13, 2005, the Examiner rejected claims 1-17 under 35 USC §101 because the claims “recite no clearly defined practical application of the claimed method or do not draw a conclusion as to the final end result of testing a software module with parameter conditions.” This point was discussed during the interview of April 8, 2005 and addressed in the response filed May 13, 2005. Claim 1 is drawn to a “method of generating a list of parameter value combinations to test” and claim 14 is drawn to a “method of generating a list of parameter values to test.” Claims 1 and 14, as well as the claims that depend from claims 1 and 14 do not include elements that require testing a software module. As discussed in paragraph 3 of the present specification:

Existing systems and methods for selecting parameter values and combinations of parameter values can be time consuming and error prone. For example, manually selecting parameter values and combinations of parameter values can take a long time. And, for complex software modules that have a large number of parameters, the manual selection of parameter values and combinations of parameter values can result in a testing procedure that does not include testing critical inputs.

Claims 1-15 include elements for generating parameter values and parameter value combinations that will ultimately be used when testing a software or hardware module. The

concrete and tangible result produced by the method of claim 1, for example, is “parameter value combinations that result in the combination function exceeding a predetermined probability value.” The concrete and tangible result produced by the method of claim 14, for example, is “parameter values that result in the probability function exceeding a predetermined probability value.” The Applicant respectfully submits that the assertion provided in the Office Action of July 13, 2005 that the claims “do not draw a conclusion as to the final end result of testing a software module with parameter conditions” does not somehow result in the claims covering non-statutory subject matter.

At the top of page 4, the Office Action alleges that the methods claimed in claims 1-15 “merely solves a model mathematical problem without limitation to a practical application.” Again, the Applicant respectfully disagrees. Claim 1, for example, includes the concrete element of “(a) providing to a user a graphical user interface that includes at least two adjustable probability curves that allow the user to graphically indicate the importance of values of at least first and second parameters.” Claim 14, for example, includes “(a) providing to a user a graphical user interface that includes an adjustable probability curve that allows the user to graphically indicate the importance of values of a parameter.” It is unclear to the Applicant how such steps can be considered to be the manipulation of an abstract idea. A specific graphical user interface is being presented to a user. Moreover, in contrast to merely involving solving “a model mathematical problem without limitation to a practical application” claims 1-15 result in specific parameter values and parameter value combinations that may be used to test a software or hardware module.

On page 4, the Office Action of July 13, 2005 next alleges that claims 1-17 “recites [sic] signal analysis that is not tied to physical structure for . . .” performing several of the claim

elements. Claims 1-15 are method claims that include definite elements and not signal analysis. The Office Action has not cited any authority, and the Applicant respectfully submits that none exists, for the proposition that elements of a method claim must be tied to a physical structure.

Based on the forgoing, the Applicants requests reversal of the rejection of claims 1-15.

**Claims 16-17**

In the Office Action of July 13, 2005, the Examiner rejected claims 16-17 under 35 USC §101 because the claims “recite no clearly defined practical application of the claimed method or do not draw a conclusion as to the final end result of testing a software module with parameter conditions.” Claim 16 is drawn to “a method of testing a software module with parameter combinations.” The method includes displaying specific information, such as “a list of parameter combinations,” “an input icon,” and “an output of the software module.” Claim 16 further includes the concrete element of “receiving an indication from a user to drag at least one of the parameter combinations to the input icon.” The Applicant respectfully submits that claims 16-17 relate to “testing a software module with parameter combinations” which is a clearly defined and practical application. The final result of testing is displayed “in a third region of the display,” such as region 708 shown in Figure 7. Claims 16-17 cover the use of a specific graphical user interface when testing and do not cover a specific testing algorithm. Specific testing algorithms and methods were well known and widely available to those skilled in art. As such, they were not required to be described and were not described in the present application.

On page 4, the Office Action of July 13, 2005 next alleges that claims 16-17 “recites [sic] signal analysis that is not tied to physical structure for . . .” performing several of the claim elements. Claims 16-17 are method claims that include definite elements, such as receiving and

displaying and are not signal analysis claims. Claims 16-17 also include the structural elements of “a display and a user selection device.”

Based on the forgoing, the Applicants requests reversal of the rejection of claims 16-17.

**B. Claims 1-17 Comply with the Written Description Requirement of 35 USC §112, First Paragraph**

Claims 1-17 have never been amended. As indicated in MPEP §2163 “Consequently, rejection of an original claim for lack of written description should be rare.” The Office Action has failed to provide any explanation of why the present claims fall into this rare situation. In rejecting claims 1-17, the Office Action merely indicates the following:

In claim 16, it is not understandable what is meant by “receiving an indication from a user”, since there is no clear and specific indication disclosed in Applicant’s Specification.

As an initial matter the Applicant would like to note that no basis for rejecting claims 1-15 has been provided and the Applicant is left in the inappropriate situation of speculating on the basis for rejecting these claims. Moreover, the Examiner appears to be rejecting claim 16 under of 35 USC §112, first paragraph by alleging that the indicated claim element is indefinite. The Applicant will address the definiteness of “receiving an indication from a user,” but respectfully submits that the rejection of claims 1-15 is insufficient and the rejection of claims 16-17 applies the wrong legal standard. As such, the rejection of claims 1-17 should be withdrawn.

Paragraph 29 of the present application is quoted below:

Figure 7 illustrates a graphical user interface 700 that may be used to test a software module, in accordance with an embodiment of the invention. User interface 700 includes a first region 702 that displays an execution matrix of parameter combinations. An input icon 704 is displayed in a second region. A software module 706 may be represented in another region. A user may test a particular parameter value combination by selecting that combination from the execution matrix and dragging that combination to input icon 704. After the

software module has operated on the parameter value combination, the results of the operation may be displayed in an output region 708.

After reading the description above and viewing Figure 7, one skilled in the art will appreciate that a user may use a pointing device, such as a mouse (shown as element 102 in Figure 1 and described in paragraph 16) to “select” a combination and “drag” the combination. As stated in paragraph 16 “[a] user can enter commands and information into the computer 100 through input devices such as a keyboard 101 and pointing device 102.” In fact, Figure 7 shows a cursor being used to drag the combination that consists of 2, 3 and 4 to input icon 704. When a user uses a pointing device to select and drag a parameter combination, a software application or computer device would receive an indication from a user to drag at least one of the parameter combinations to the input icon.

Based on the forgoing, the Applicant respectfully submits that “receiving an indication from a user to drag at least one of the parameter combinations to the input icon” complies with the written description requirement of 35 USC §112, first paragraph and is additionally definite and in compliance with 35 USC §112, second paragraph. Moreover, the elements in claims 1-15, which have never been amended, are all specifically described in the specification and drawings and in compliance with 35 USC §112, first paragraph.

The Applicant accordingly requests reversal of the rejection of claims 1-17 under 35 USC §112, first paragraph.

**C. The “to test” Language Used in Claims 1-15 is Definite and Complies with 35 USC §112, Second Paragraph**

On page 5 of the Office Action of July 13, 2005, the Examiner alleges that the use of “to test” language included in the preambles claims 1 and 14 renders claims 1 and 14 indefinite because “[i]t is not clear to the Examiner what test is intended.”

As stated above, claim 1 is drawn to a “method of generating a list of parameter value combinations to test” and claim 14 is drawn to a “method of generating a list of parameter values to test.” The language “to test” merely indicates what the selected parameter values and parameter value combinations may ultimately be used for testing. Claims 1 and 14 do not include any elements that require the actual testing. The methods of claims 1 and 14 are performed before a software or hardware module is tested. After the methods of claims 1 and 14 are performed, and as stated in paragraph 28 of the present specification “[a]ny one of the well known software module testing methods may be used.”

The Applicant accordingly requests reversal of the rejection of claims 1-15 under 35 USC §112, second paragraph.

**D. The use of “Parameter” in the Specification and Claims Has a Definite Meaning to One Skilled in the Art**

---

As stated previously, the Applicant respectfully submits that after reading the present specification, the meaning of “parameter” would be clear to one of ordinary skill in the computer programming art. For example, as stated in paragraph 2 of the present specification:

Software modules, such as application programming interfaces, continue to become increasingly complex. Such modules may include a larger number of input parameters and parameter combinations. The testing of software modules typically includes selecting parameter values and parameter value combinations. The combinations of parameter values are then applied to the software module and the resulting output is analyzed.

The above description indicates that parameters are input and output arguments used by software modules, such as application programming interfaces.

On pages 2 and 5-6 of the Office Action of July 13, 2005 the Examiner objects to the use of “parameter” in the specification and claims because “Applicant did not provide a clear and specific definition in the Specification disclosed for the Examiner to understand what is meant by

'parameter' in applicants' claimed invention." On page 5 the same Office Action then appears to agree that "parameter" has a well known meaning. But, then later on page 5 the same Office Action then appears to suggest that the Applicants "do not have any further information for providing essential structural connections between testing hardware and software and the importance of a parameter."

The language used on page 5 of the Office Action of July 13, 2005 is cited below:

Applicants argue that parameter has a widely well-known meaning to those of ordinary skill in the computer programming art. For example, the following figure and description is from U.S. Patent No. 6,714,952 (See Figure 8 and Col. 11, line 55 - Col 12, line 5.). The Examiner agrees with Applicants. it is well known in the art that parameters are input and output arguments used in the software programming language; however, the Applicants do not have any further information for providing essential structural connections between testing hardware and software and the importance of a parameter. In addition, how on [sic] skill [sic] in the art can see Figure 8 and col. 11, line 55-col. 12, line 5 described and depicted in the U.S. Patent No. 6,714,952 if Applicants do not provide such information in the BACKGROUND.

As an initial matter the Applicant respectfully submits that U.S. Patent No. 6,714,952 merely evidences the fact that the definition of "parameter" was notoriously well known in the art and the use of "parameter" in the present application and claims is consistent with that well known meaning. It also appears that the Examiner agrees. Under the proper legal standard the reader of the present specification is presumed to be one skilled in the art and therefore there is no reason or requirement to cite U.S. Patent No. 6,714,952 in the Background.

In the citation provided above, the Examiner appears to indicate that the Applicant is required to provide "essential structural connections between testing hardware and software and the importance of a parameter." The Examiner has not provided any authority for this requirement. Moreover, at most the claims refer to the importance of "values" of parameters and not the importance of parameters. As stated in paragraph 22, of the present application "The

relative interest of a particular parameter value may be the result of numerous factors determined by a user. For example, a high interest parameter value may be a value that is likely to occur when a software module is in operation or a critical value.” It is clear from this description and the rest of the application that a user decides which parameter values are important. A parameter value may be important if it is likely to occur or is a critical value. For example, a user may know that a particular software module has an input parameter of “x” and “x” will frequently have a value of “5” when the software module is operating. This could be determined by considering the intended use of the software module. Based on this information, a user would determine that a value of “5” for parameter “x” is important.

Based on the forgoing, the Applicant requests reversal or withdrawal of any rejection or rejection based on the use of “parameter.”

**CONCLUSION**

The rejection contained in the Action of July 13, 2005 should be reversed for at least the reasons recited above. Reversal of the rejections is requested.

Respectfully submitted,

Date: December 13, 2005

*William J. Miller 51,393*

Charles L. Miller  
Registration No. 43,805  
BANNER & WITCOFF, LTD.  
10 S. Wacker Drive  
Suite 3000  
Chicago, IL 60606-7407  
Telephone: 312-463-5000  
Facsimile: 312-463-5001

**CLAIMS APPENDIX**

1. A method of generating a list of parameter value combinations to test, the method comprising:

- (a) providing to a user a graphical user interface that includes at least two adjustable probability curves that allow the user to graphically indicate the importance of values of at least first and second parameters;
- (b) converting the probability curves into probability functions;
- (c) combining the probability functions into a combination function; and
- (d) selecting parameter value combinations that result in the combination function exceeding a predetermined probability value.

2. The method of claim 1, wherein the combination function is equal to the product of the probability functions.

3. The method of claim 1, wherein the combination function is normalized over the definition domains and is equal to:

$$\sum_{i=1}^n (2 * P(x_i) * \text{Max}(P(x_i)) - P(x_i)^2)$$

where  $n$  is the number of probability functions,  $P(x_i)$  is the probability function for parameter  $x_i$  and  $\text{Max}(P(x_i))$  is the maximum value of the  $P(x_i)$  probability function.

4. The method of claim 1, wherein the graphical user interface in (a) allows the user to select domains for the values of the at least first and second parameters.

5. The method of claim 4, wherein a domain is selected by providing minimum and maximum values.

6. The method of claim 1, wherein (b) comprises performing polynomial curve fitting.

7. The method of claim 6, wherein the grade of the polynomial corresponds to a desired accuracy level.

8. The method of claim 1, wherein the parameters are numerical variables.
9. The method of claim 1, wherein the parameters comprise string parameters.
10. The method of claim 9, wherein a string parameter comprises length.
11. The method of claim 1, wherein the parameter combinations comprises inputs to a software module.
12. The method of claim 11, wherein the software module comprises an application programming interface.
13. The method of claim 1, wherein the parameter combinations comprise inputs to an integrated circuit.
14. A method of generating a list of parameter values to test, the method comprising:
  - (a) providing to a user a graphical user interface that includes an adjustable probability curve that allows the user to graphically indicate the importance of values of a parameter;
  - (b) converting the probability curve into a probability function; and
  - (c) selecting parameter values that result in the probability function exceeding a predetermined probability value.
15. The method of claim 14, wherein the probability function is a continuous function and (c) includes selecting parameter values from a group of discrete parameter values that result in the probability function exceeding a predetermined probability value.
16. In a computer system having a graphical user interface including a display and a user selection device, a method of testing a software module with parameter combinations, comprising:
  - (a) displaying in a first region of the display a list of parameter combinations;
  - (b) displaying in a second region of the display an input icon;
  - (c) receiving an indication from a user to drag at least one of the parameter combinations to the input icon; and

(d) in response to (c) displaying in a third region of the display an output of the software module.

17. The method of claim 16, wherein the list of parameter combinations includes parameter combination usage probabilities.

**VIII. EVIDENCE APPENDIX**

None.

**IX. RELATED PROCEEDINGS APPENDIX**

None.